UNIVERSIDAD DEL NORTE

Retrieving, Annotating and Recognizing Human Activities in Web Videos

by

FABIAN DAVID CABA HEILBRON

A thesis submitted in partial fulfillment for the
degree of Master of Science

in the

Department of Electrical and Electronic Engineering

January 2014

UNIVERSIDAD DEL NORTE

Abstract

Department of Electrical and Electronic Engineering

Master of Science

by FABIAN DAVID CABA HEILBRON

Recent efforts in computer vision tackle the problem of human activity understanding in video sequences. Traditionally, these algorithms require annotated video data to learn models. In this work, we introduce a novel data collection framework, to take advantage of the large amount of video data available on the web. We use this new framework to retrieve videos of human activities, and build training and evaluation datasets for computer vision algorithms. We rely on Amazon Mechanical Turk workers to obtain high accuracy annotations. An agglomerative clustering technique brings the possibility to achieve reliable and consistent annotations for temporal localization of human activities in videos. Using two datasets, Olympics Sports and our novel Daily Human Activities dataset, we show that our collection/annotation framework can make robust annotations of human activities in large amount of video data.

We investigate the performance of existing approaches in our novel dataset of natural human activities. Unfortunately, we note that existing methods can not tackle the noisy nature of web videos. For instance, dense point trajectories (one of the most popular feature extraction method) can be heavily corrupted when applied to videos acquired with moving cameras. In this work, we explore the use of weak video stabilization to compensate for coarse camera motion and to isolate the subtle motions of interest that better represent the events in the sequence. Instead of stabilizing the entire sequence, our weak stabilization operates on local time ranges using a temporal sliding window. Our algorithm computes trajectories over the locally stable windows, which result in robust trajectory estimation and improved feature descriptors. Our experiments on four benchmark datasets (Hollywood 2, Olympic Sports, HMDB51, Daily Human Activities dataset) show action recognition performances that improve comparable state-of-the-art algorithms.

Acknowledgements

First of all, I would like to thank my advisor and mentor, Professor Juan Carlos Niebles, for providing a never ending support. His constant dynamism and motivation makes working with him an exceptionally experience. I would also like to thank my colleagues at the VisionLab, for lots of stimulating talking about computer vision, and the thoughts about all other important aspects of life. Finally, I owe great thanks to my friends for their laughter and my family for their endless support.

Funding for this work was provided by a COLCIENCIAS National Young Scientist and Innovator Fellowship, a Microsoft Research Fellowship and an Stanford Fellowship.

Contents

List of Figures

List of Tables

Dedicated to my family.

Chapter 1

Introduction

"In God we trust. All others must bring data."

W. Edwards Deming

Keywords: Computer vision, Human activity recognition, Video annotation, Crowdsourcing, Video stabilization

We are in the midst of data revolution where visual content has a protagonist role. For instance, YouTube reports that around hundreds hours of video are uploaded each minute to their servers. Uploaded content ranges from a view of astronauts walking in the space to the first steps of a baby at home. This large amount of data opens new opportunities for computers to acquire knowledge about human activities.

The ability to automatically understand and recognize human activities, brings possibility to develop applications such as: video surveillance systems capable to alert suspicious activities, novel ways of human computer interactions for gaming, monitoring performance in sports, automated household assistants and indexing content in web platforms.

Nevertheless, state-of-the art activity recognition algorithms are still far from achieving high accuracy to recognize the large visual space of human activities. One of the key problems of these approaches is due to the small number of well sorted and labeled video datasets for training, testing and evaluation of algorithms. Existing datasets are either too small in the number of categories (in the order of 100) or the number of samples per category (in the order of 100).

Our work focuses on studying scalable ways to collect and annotate human activity data from web videos. We design a framework to collect and annotate a dataset of

natural human activities. We then investigate the performance of current state-of-the-art methods to recognize those high level activities (rather than the traditional kinematic action categories). Finally, several challenges such as camera movements are tackled in order to obtain an enhanced description of the human movements. For this, we introduce a novel feature extraction approach capable of overcoming undesired camera motions.

1.1 Goals

Our basic objective is to build tools to gather videos depicting human activities and annotate them with both high quality and scalable speed. We aim at designing a crowd sourced framework to collect and curate human activities in large scale video collections. We seek to cultivate a new era of advances in human activity recognition problems by gathering and annotating a rich new collection of high level activities. We plan to publish an API to allow for easy programmable access to the collected data and tools through common software interfaces like Matlab and Python.

1.2 Contributions

We propose a novel framework to collect and annotate human activities from web videos. We construct a novel large scale video database of high level human activities (i.e. shoveling snow) structured using a hierarchical taxonomy. We employ a subset of our novel database to evaluate state-of-the-art algorithms for action recognition. Moreover, we introduce a simple approach to stabilize visual features extracted from the video sequences. We summarize our contributions as follow:

We introduce a novel framework to collect and annotate human activities from web media servers such as YouTube. To collect data, we propose a harvest method to retrieve candidate videos related with an intended activity. Afterward, we rely in the power of human crowds to curate all retrieved videos. We demonstrate that our framework successfully retrieve human activities from YouTube videos.

In order to overcome the limitations of the existing datasets for training, testing and evaluation human activity recognition algorithms, we introduce a large scale video database of high level activities. Our novel database contains 75 different classes with a total of 46 leaf categories with around 100 video samples. We provide temporal boundaries where the high level activity occurs. Furthermore, we provide a hierarchical taxonomy that structures the collected activity in a semantic sense.

The final contribution of this thesis is the evaluation of different state-of-the art algorithms for activity recognition. First, we evaluate the algorithms in a subset of our novel database of high level human activities. We find that existing approaches has a low performance. We attribute this to the high diversity of our database and the challenges related to the noisy capture condition of web videos such as camera movements. To overcome the camera motion, we propose a simple approach to stabilize visual features extracted from the videos.

1.3 Thesis overview

The remainder of this thesis is organized in the following way. In Chapter 2, we review previous related work for four main topics: video annotation, human activity recognition, datasets and benchmarks and video stabilization. Chapter 3 describes three contributions: (1) our novel framework to collect and annotate human activities, (2) our novel database of natural human activities, and (3) evaluation of two different algorithms for action recognition in our novel dataset. Chapter 4 then proposes a novel feature extraction approach capable of filtering coarse motions while preserving subtle motions of interest. Finally, Chapter 5 provides concluding remarks.

Chapter 2

Related Work

In this chapter, we first provide a brief review of existing video annotation tools and introduce the crowdsourcing paradigm in Section 2.1 (we stress the uncover ability to annotate human activities with both high accuracy and scalable speed). We then look human activity recognition algorithms in Section 2.2. Next, we review existing datasets in which human activity algorithms are benchmarked (Section 2.3). Finally, we study different approaches for both video stabilization and robust feature extraction in videos with noisy camera motions (Section 2.4).

2.1 Video annotation

In this section we overview some of the most relevant previous work on the topics of crowdsourcing and video annotation tools. First, we review the most popular approaches in which crowdsourcing is involved. We then describe existing approaches for video annotation.

2.1.1 Crowdsourcing

Crowdsourcing can be defined as a collaborative participation in which a crowd of people helps to solve problems. Typically, crowdsourcing involves a reward; sometimes associated with money, public acknowledgment, or simply entertainment [24]. There is a large amount of applications in which crowdsourcing is involved. From book digitalization [58] to prediction of protein structures [11], crowdsourcing is used to solve several problems that help computers to acquire knowledge.

Recently, crowdsourcing became a new trend in computer vision. Specifically, Amazon Mechanical Turk (AMT) serves as an inexpensive platform to label visual data accurately [60]. AMT is an online platform in which a crowd of workers, sometimes called **Turkers**, are waiting to solve simple tasks for a micro reward. In this scenario, a **Requester** send a **Human Intelligence Task (HIT)** to AMT. Then, a Worker solves the HIT and waits for a Reward. Since AMT is an open-access platform, it can be seen as an hostile scenario, due to a possibly large amount of malicious workers.

The challenges of getting high quality annotations at low cost are widely investigated [1, 20, 25, 26, 29, 48, 55]. Ipeirotis et al. [26], investigate techniques to accurately estimate the quality of worker annotations. They propose several strategies such as Gold standard data and multiple labels for the same task to allow the rejection and blocking of the malicious workers. Raykar et al. [48], propose a method to rank annotators in order to achieve a maximization in annotation quality. Other studies such as [20] and [55], investigate major voting or consensus strategies in order to obtain reliable labels.

2.1.2 Annotation tools

Several researchers have designed tools for annotating videos. For instance, Mihalcik and Doermann [43] propose ViPER, an off-line user interface to provide spatial annotations in video sequences. Ali et al. [3] introduce Flowboost, a sparse labeling technique to annotate videos from key frames. Yonemoto [64] presents a video annotation tool of 3D videos. Dollar et al. [14] design an intuitive user interface to annotate pedestrians in video sequences. They use a soft-labeling technique in order to reduce the human effort.

With the discovery of crowdsourced labeling, novel applications emerged to annotate video data. Vondrick et al. [59] present VATIC, an open platform to label at low cost and high quality objects in video sequences. Inspired in [57], VATIC explores balancing computer and human effort in video annotation. LabelMe Video [66] is another crowdsourced annotation tool for video annotation. In contrast to VATIC [59], they allow free polygonal paths annotation (VATIC only allow bounding boxes annotations).

A few studies have been conducted in the specific field of human activity annotation. Fisher et al. [17] propose, explore the labeling of human activities. These annotations include bounding boxes around humans and description of their movements. Laptev et al. [35] use movie scripts to annotate human activities in long video sequences (Hollywood movies). Recently the work in [45], explores human activity annotation using crowdsourcing. They use collaborative and individual filtering strategies to annotate temporal boundaries of actions in video sequences.

Previous work focuses on the video annotation stage. In this work, we investigate the overall process of collecting and annotating human activities in videos. In contrast to existing approaches, our collection stage overcomes several challenges to reduce the amount of noisy videos. Moreover, we study the annotation of human activities at a longer time scale, including more complex activities such as "brushing teeth" and "mixing drinks", among others, rather than annotating actions such as "open" or "walk".

2.2 Human activity recognition

Human action recognition is a main topic in computer vision. This aims to analyze, recognize, detect and automatically infer human behavior in video sequences. Previous works in this field show satisfactory results on different datasets [31, 50, 53, 61]. However, these datasets do not represent neither scale nor diversity the visual world of human actions.

Davis and Bobick [5] introduce one of the first method for human action recognition. Their method represents actions as spatio-temporal volumes templates. For classification, their approach calculates HU moments [**?**] to describe the visual appearance of human actions in the videos. However, this method fails in video sequences where background subtraction is difficult, there is camera movement or actions are very complex. Efros et al. [15] present a motion descriptor based on optical flow measurements in spatio-temporal volumes for each stabilized human figure, by associating a correlation measure on the nearest neighbors classification scheme. This method requires strong supervision due to required annotations in each resulting spatio-temporal volume. Recently, ActionBank [50] proposes a high-level representation for videos, which encodes semantic information. There, an action bank is used as a detector, thereby, a video is represented with an action template correlation scheme in the bank. The collection of the action bank is invariant to appearance, scale, viewpoints and action execution pace changes.

Another approach used for action recognition tackles video representation as local patches in space and time. A popular current trend in action recognition methods relies on the bag-of-words model to represent descriptors obtained from interest points, followed by discriminative or generative methods of machine learning. Laptev [33] introduces the Harris3D feature detector, which is an adaptation of Harris corner detector to spatio-temporal domain. Its interest points operator detects local structures where each pixel image value have significant variations in space and time. In a later work Laptev et al. [34] propose HOG and HOF descriptors, in order to characterize local movements and appearance for human action recognition, by calculating histograms of spatial gradients

and cumulative optical flow in the neighborhood of the interest points thrown by the Harris3D detector. These approaches follow the bag-of-words model. This method requires the construction of a dictionary of visual words and its quantization to generate histograms. Then, recognition is performed using a Support Vector Machine (SVM) [10]. Dollar et al. [13] compare local descriptors in terms of brightness, gradients and optical flows from the image. Their method captures local regions containing complex patterns of movements, including significant changes in space and time in the values of the image. Despite the promising results of these approaches, the bag-of-words model fails to capture spatio-temporal relationships that are essential to recognize more complex actions or activities.

Tracking body parts and representing them by motion trajectories is another scheme traditionally used for action recognition. This approach requires high precision algorithms for tracking people, which is a really difficult task to perform due to the complexity of the videos where there is usually occlusion, multiple people, no relevant information, camera movement and other disturbances. For instance, Ali et al. [4] introduce a method for action recognition which characterize the nonlinear dynamics of human actions using the theory of chaotic systems. Fanti et al. [16] propose to detect and follow the features points frame per frame. This method combines multiple cues, such as position, speed and appearance in the learning and detection phases. Wang et al. [61] propose a dense feature points tracking to perform action recognition in wild videos. Their method computes local descriptors in order to obtain visual cues about human activities. Wu et al. [63] use Lagrangian particle trajectories which are dense trajectories obtained by optical flow throughout the temporal domain.

2.3 Datasets and benchmarks

Early human action video data sets like KTH [51] and Weizmann [21] present humans performing simple and distinct actions like walking and jumping jacks; the videos are low resolution with mostly static backgrounds, little clutter, and easily segmentable humans. Numerous methods for action recognition have report high accuracy on these datasets [50, 53].

Subsequent datasets (see Table 2.1) relax the environment assumptions leading to more challenging recognition tasks with difficult background and camera angles. UCF Sports [49] and Olympic Sports [46] increase the action complexity by focusing on highly articulated sporting activities; UCF YouTube [37] increase the data set size from hundreds to thousands of samples; HOHA1 [35] and HOHA2 [40] move beyond sporting actions into everyday actions like hand-shakes and answer-phone.

Datasets	Classes	Clips	Description
KTH [51]	6	600	Kinematic actions staged by amateur actors.
Weizmann [21]	10	90	Kinematic actions staged by amateur actors.
UCF Sports [49]	9	182	Sport related movements from TV and movies.
Hollywood2 [40]	12	1707	Daily actions from Hollywood movies.
Olympic Sports [46]	16	792	Olympic sports related movements collected from YouTube.
HMDB51 [32]	51	6766	Action collection retrieved from several source of web content.
UCF101 [52]	101	13320	Largest human action dataset. It is collected from YouTube.

TABLE 2.1: Properties of existing datasets for action recognition. The column "Clips" refers to the total videos in the dataset.

To the best of our knowledge, the largest existing datasets for activity recognition are UCF101 [52] and HMDB51 [32]. Both data sets compile YouTube videos and have more than 50 categories. These more recent, larger data sets indeed present a greater challenge than the earlier, so-called kinematics data sets [36] like KTH [51]. However, they are composed by simple atomic actions (do not include high level activities or events), such as walking, jogging, running, among others.

2.4 Video stabilization

A large amount of work has studied the problem of human action recognition in videos [2]. In this section we overview some of the most relevant previous work on the topics of video stabilization and video feature extraction.

A common methodology for video stabilization relies on estimating the global camera motion. One approach for this estimation computes sparse visual features [6, 8, 68] such as corners [9] and estimates a warping matrix between consecutive frames. Others prefer to use all pixels in the image to compute an alignment [39, 42], but tend to suffer under-fitting due to local outliers. An alternative methodology defines a model for camera motion [7, 23] and uses multiple frames to estimate its parameters. Unfortunately, there is a large variation in camera motions and it is difficult to capture them in a single model. Once the camera motion is estimated, most algorithms use it to perform image alignment or warping [54]. Unfortunately, warping usually introduces empty image regions in the aligned image. These areas may be recovered using impainting methods [62] at a high computational cost. Finally, instead of fully stabilizing sequences, [19, 22] propose to simulate professional camera motion in videos taken with hand-held cameras. Unfortunately, not all camera motion is removed and the application of these methods to action recognition is limited. Most related to our approach (see Chapter 4), Park et al.

[47] recently show how the use of weak video stabilization based on a coarse optical flow can lead to improved pedestrian detection in videos. Their goal is to isolate limb motion while canceling pedestrian translation and camera motion. In this work (in Chapter 4), we explore the extension of this technique and its applicability to feature extraction for action recognition.

In another line of work, researchers have studied the issue of extracting video features for recognition that are robust to camera motion [27, 31, 63]. When applied to videos with large camera movement, traditional video feature extraction methods tend to generate a large number of features that are mostly related to the camera motion [13, 33, 61]. In order to overcome this issue, Wu et al. [63] propose the use of Lagrangian particle trajectories for action description in videos acquired with moving cameras. Their method compensates for the global camera motion and only extracts features that exhibit motion independent to the camera movement, outperforming traditional feature extraction algorithms. Matikainen et al. [41] present a technique for action recognition with quantized trajectories of tracked features. More recently, Wang et al. [61] present a method for action recognition using dense sampling of point trajectories. Their method handles large camera motions by limiting the maximum length of tracked trajectories. In spite of their simplicity, these dense trajectory features achieved state-of-the-art performance in benchmarking datasets. In order to improve upon these dense trajectories, Jain et al. [27] propose a method to estimate more reliable motion features for action recognition. Their method obtains improvements on feature robustness by first decomposing optical flow into dominant and residual motions. Dominant motion is estimated using an affinity model and subtracted from the computed optical flow to obtain the residual motion. This information is then used to compute local motion descriptors. While the method is simple and improves recognition performance, residual and dominant motion estimations are not reliable when the dominant motion is related to the actor.

We address some of the limitations of current methods by introducing the use of weak stabilization to improve robustness of dense trajectories in videos with large camera motion. We introduce the details of our framework in Chapter 4.

Chapter 3

Retrieving and Annotating Human Activities From the Web

With the growth of on-line media, surveillance and mobile cameras, the amount and size of video databases increase at an incredible pace. For example, YouTube reported that over 100 hours of video are uploaded every minute to their servers [65]. Arguably, people are the most important and interesting subjects of such video. Under this perspective, recognizing human activities and actions is crucial for building smarter computer vision systems, semantically aware video indexes and more natural human-computer interfaces. However, despite the explosion of video data, the ability to automatically recognize and understand human activities is still rather limited. Challenges related to large variability in execution styles, complexity of the visual stimuli in terms of camera motion, background clutter and viewpoint changes, as well as level of detail and number of activities that can be recognized remain unsolved. Two main important limitations are (1) that existing datasets are either too small in the number of categories (in the order of 100) or the samples per category (in the order of 100), and (2) that existing datasets are typically collected and annotated with not scalable costly manual labels.

In order to overcome these limitations, Deng et al. [12] introduce the use of Crowdsourcing platforms (Amazon Mechanical Turk) to collect and annotate a large scale hierarchical image database called ImageNet. They tackle several challenges related to harvesting, understanding, and harnessing big visual image data. In the video domain, VATIC [59] and LabelMe Video [66] begin to add crowdsourced annotations for localizing objects and attributes in video sequences. Recently, Nguyen-Dinh et al. propose a method to annotate starting and ending times of simple actions (i.e. pour, take and open) with the help of Amazon Mechanical Turk. Regardless of incremental advances in

video annotation, there is an unsolved limitation related to the collection and annotation of accurate human activity data in video sequences.

In this chapter, we introduce a framework that close the gap in activity recognition, creating a way to easily collect and annotate accurate data from web media-servers. Our idea is simple: we search the web for candidate videos using text queries. Amazon Mechanical Turk workers then verify each candidate video and determine if it matches an intended class activity. Finally, we temporally localize the activity with the help of Amazon Mechanical Turk. We show that our framework achieves a high accuracy in two different human activity recognition benchmarks (Olympic sports [46] and a novel dataset of daily activities).

3.1 Human activity annotation

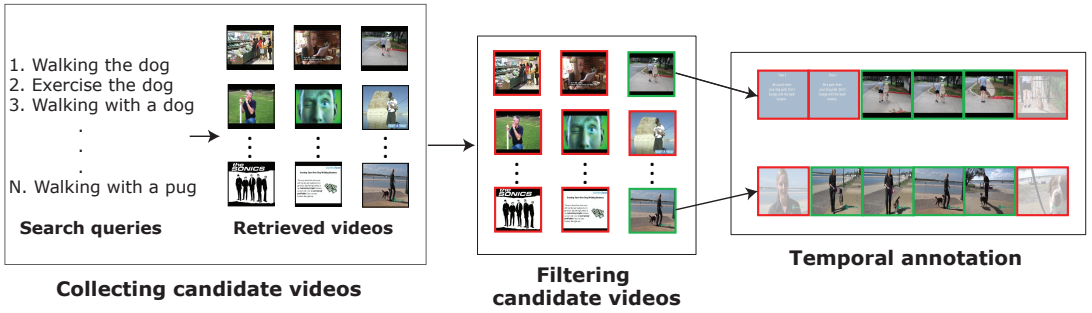

FIGURE 3.1: Human Activity Annotation Framework overview. **Left**: We search candidate videos in the web for each element in our activity list (including expanded queries, explained in Section 3.1.1). **Center**: Retrieved videos are verified by Amazon Mechanical Turk workers and clean out those videos which are not related with any intended activity (see Section 3.1.2). **Right**: Temporal localization provides starting and ending times in which activity is performed (multiple Amazon Mechanical Turk workers annotate a single video, explained in Section 3.1.3).

Our goal is to build an automatic system to retrieve videos depicting human activities. We heavily rely on the crowd and specifically, Amazon Mechanical Turk, to help acquire and annotate web videos. Our acquisition pipeline has three main steps: (1) collection, (2) filtering, and (3) temporal localization (Figure 3.1 illustrates our framework). Next, we discuss these steps in more detail.

3.1.1 Collecting candidate videos

Here we introduce the approach to collect candidate web videos that depicts a person performing an activity. In this stage we collect all video sources required for the entire

annotation pipeline. Our system requires as input a list of concepts related to the desired human activities (i.e. the name of the activity). Then, concepts are expanded with WordNet [44]; queries are expanded using hyponyms, hypernyms and synonyms, which increases both the number of retrieved videos and the variance in the visual content. For instance, for the key concept "Walking the dog", we also include queries like "Exercise the dog".

Once we have a set of queries, the system connects to on-line media sources, such as YouTube and submits all these queries. We store in a database the matching results and include all the available metadata such as author name, video description, video license and video tags.

3.1.2 Filtering candidate videos

Annotating human activities from web videos is still a difficult problem for computers. We rely on human workforce to provide accurate annotations to determine if a video matches with desired activity. With the help of Amazon Mechanical Turk (AMT) which is an useful platform to collect human annotated data, we hire humans around the world to verify all candidate videos retrieved.

To complete this hard for computers but easy for humans labor, we design an intuitive user interface (UI) that shows several videos and a question form to ask if the video is related with a desired activity. Since we focus on collecting videos of real human activities, questions such as "is this video an animation?" or "is this video captured from a video game?" are included to the UI. Figure 3.2 shows the UI designed to filtering candidate videos. Turkers can read instructions at any time by clicking this panel. Moreover, we rely on the valuable opinions of the workers and provide a feedback form to allow them to write about the task.

Since AMT is a free-access platform, several spammers and malicious workers could be found. Recent studies report that around 30% of workers are spammers [25]. In order to identify those malicious workers, we rely on **Gold Standard** data inside our task [30]. This means that we include verifiable questions into the task to avoid inaccurate annotations. Another quality assessment, is to select "Master Workers" [1]. We study these strategies in Section 3.2.1.

[1] "Masters are elite groups of Workers who have demonstrated accuracy on specific types of HITs on the Mechanical Turk marketplace.". https://www.mturk.com/mturk/help?helpPage=worker

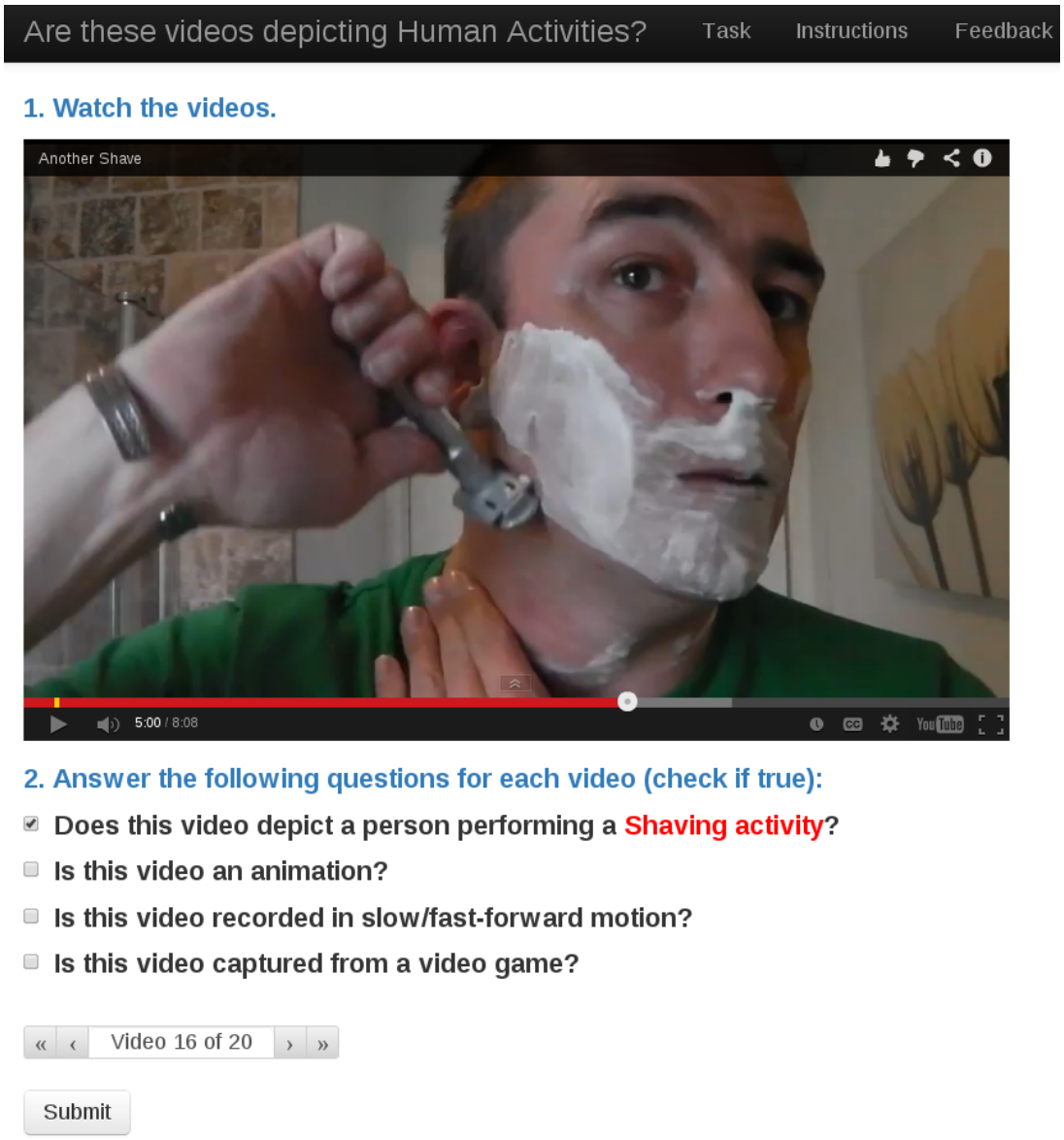

FIGURE 3.2: Screen shot of the user interface for filtering candidate videos. It is designed so that users can provide simple YES/NO questions. Moreover, we include a panel to describe the objectives of the task and the guidelines to achieve the reward. Additionally, tukers tell us their opinion about the HIT using the feedback panel.

3.1.3 Temporal annotation

Traditionally activity classification requires trimmed videos to train learning algorithms. Due to this requirement, we design a system to annotate the temporal localization of activities in video sequences. We rely again in AMT workers to identify video frames associated with a given activity. For this, turkers annotate starting and ending times of an intended activity. We design an AMT interface that allows frame navigation and selection of the temporal boundaries associated to human activities (see Figure 3.3). The UI includes detailed instructions to successfully perform the HIT.

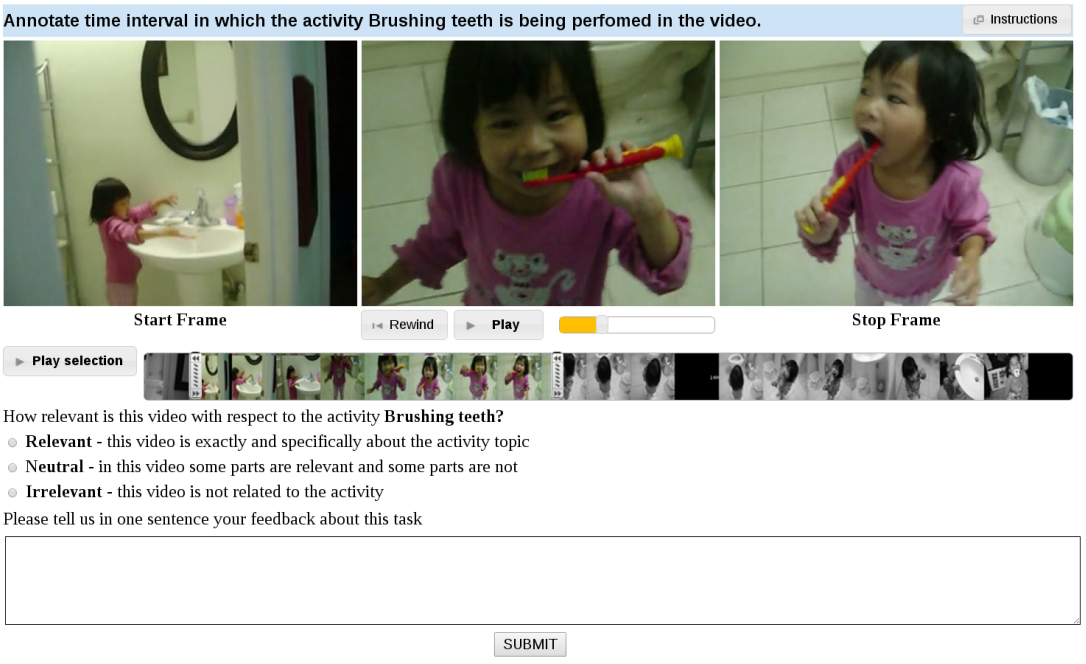

FIGURE 3.3: Screen shot of the user interface for temporal localization of human activities. It is designed so that users can annotate starting and ending times of activities using a simple slider. Users have three frames boxes where starting, current and ending frames are displayed. The interface preload the video so that users can efficiently navigate all frames. Instruction and feedback panels are included in the UI.

We employ multiple turkers to annotate a single video. In this way, we can achieve more accurate annotations and we could discover isolated annotations that can be treated as bad annotations provided by malicious workers. A complete linkage clustering (CL) is used to group and find representative annotations among the workers. CL merges clusters in order of proximity; the closest clusters are merged at first, and the furthest are joined at last. We constrain the CL algorithm using an **overlap threshold** between clusters.

Figure 3.4 shows a synthetic example of how we perform clustering over multiple annotations. We can find different groups that represent different segments in the video in

which an activity occurs. Additionally, isolated annotations can be detected if it is not merged into a representative group.

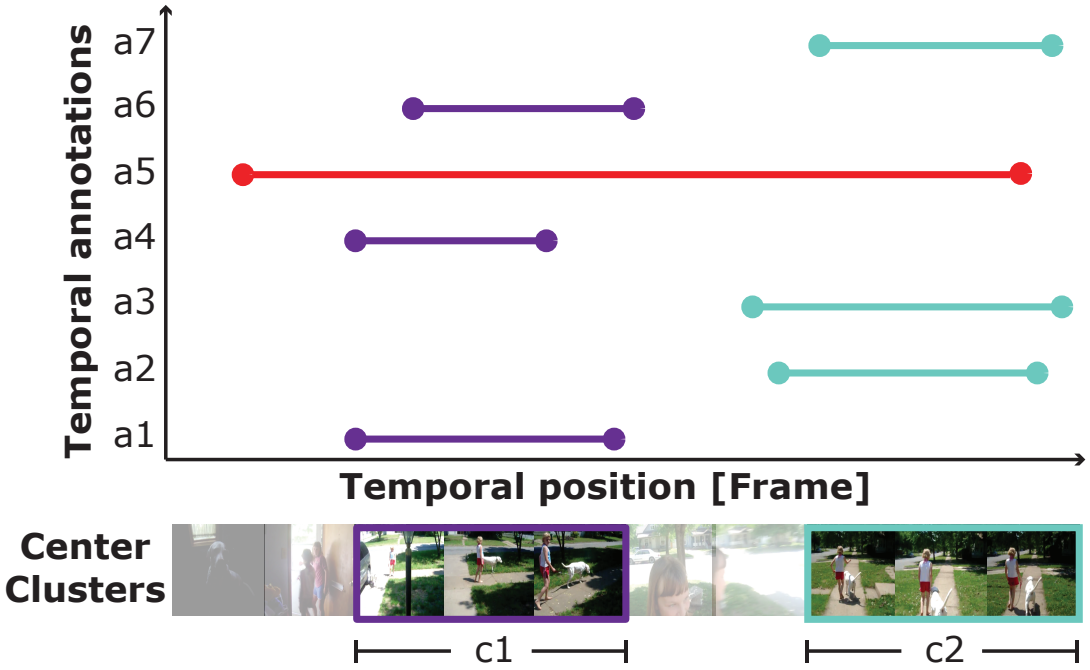

FIGURE 3.4: Temporal localization example for activity "Walking the dog". In this example, seven workers annotate the video. Then, our clustering algorithm merge consistent annotations in two representative groups. Additionally, our approach allows to find an isolated annotation (a5). (This figure is best viewed in color.)

Different from previous works, our framework includes all stages required to collect and annotate human activities. Moreover, we focus on temporally annotate web videos which usually include visual content beyond the activity of interest, which tends to be confined to a shorter time interval within the sequence.

3.2 Experiments and results

3.2.1 Constructing a daily human activities benchmark

We construct a novel dataset of daily human activities. Annotations in this dataset are provided manually (filtering and temporal localization). We populate the activity list using high levels daily human events. Our novel dataset contains 10 different classes and around 100 samples per class. Figure 3.5 shows random frames for each category in the introduced benchmark. Next, we evaluate the full collection/annotation process of our novel benchmark and investigate the effect of different parameters involved.

FIGURE 3.5: Random frames of our novel dataset of 10 daily activities. Our dataset contains 100 sample videos from each of 10 activities: brushing hair, brushing teeth, ironing clothes, painting, preparing salad, putting in contact lenses, shaving, shoveling snow, vacuuming floor, walking the dog. Our dataset arose several challenges as camera movements, occlusion, dynamic background, etc. Moreover, our novel dataset is composed by high level activities which represent a major dare to activity recognition algorithms.

3.2.1.1 How to diversify retrieved video content?

If you go to YouTube and type in the search bar the query "Preparing drinks", you find several tutorial style videos. Moreover, we find that a considerable amount of videos are near-duplicates. We employ two simple strategies to diversify the retrieved content. We group automatically the content for each activity class using the video author information. Then, we compute a pairwise similarity measure across all videos of the same author using video key frames as in [**?**]. If a high similarity is found, we filter this record in the database. Second, we use the category information provided by YouTube to enlarge the variance of the content. Figure 3.6 shows the category distribution before and after our diversification strategy. We observe that initial distribution are biased (**Left** sub-figure) with mostly "How to & Style", "People & Blogs" and "Sports" related videos. In order to get a uniform distribution, we limit the number of retrieved videos for each category. Figure 3.6 (**Right** sub-figure) illustrates that category distribution are normalized which represent better variety on the retrieved content.

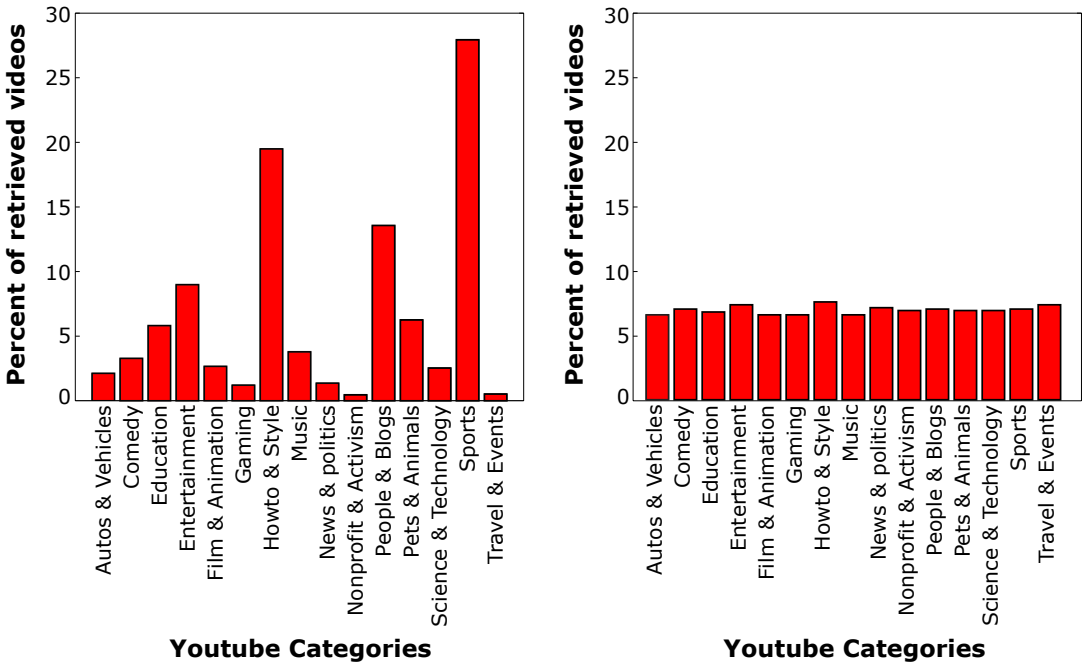

FIGURE 3.6: Obtained YouTube category distribution from candidate videos. We investigate the distribution of YouTube video category, in order to identify biases in collected data. **Left** sub-figure shows the distribution computed before our simple diversify strategy. **Right** sub-figure shows that our method brings a more uniform distribution over YouTube categories.

3.2.1.2 How much Gold Standard data to filter candidate videos?

We investigate how much Gold Standard data is required to obtain high quality annotations for the task of filtering candidate videos. To collect our novel dataset, we include 20 videos in each HIT (filtering stage). In this experiment, we compute the precision and recall in terms of the number of seeds (Gold Standard data) planted for each HIT. Precision is measured as the total true positive videos containing a desired activity divided by the sum of both true positive and false positive videos retrieved. Otherwise, recall is computed as fraction of videos containing an intended activity from the total videos on the Ground Truth. Finally, we report F1-score (henceforth F-score) in order to consider both precision and recall in our evaluation. Figure 3.7 shows the F-score (measure of test accuracy, computed based on precision and recall) varying the number of seeds in a range of [0-10]. We observe that annotation accuracy stabilizes when five or more seeds are planted in the HIT. We attribute this stabilization to that answers from malicious workers are filtered (if a worker provides a bad answer for a gold standard data, the worker will not be able to submit the task).

3.2.1.3 Do expert workers improve the quality of annotations?

To evaluate if expert AMT workers perform more accurate annotations (filtering stage), we rely on a turkers categorization provided by AMT called "Master Worker". Figure 3.7 compare the F-score obtained by Non-Master Workers vs Master Workers. As mentioned above for Non-Master Workers around 5 seeds are required to get high accurate annotations. However, in the case of Master Workers, we observe that without planted seeds, we achieve a high F-score. Moreover, we observe that accuracy becomes independent of the number of seeds. In general, we note that "Master Workers" perform significantly more accurate annotations that "Non-Master Workers".

3.2.1.4 How many workers to annotate a single video?

To annotate starting and ending times of activities, we rely only on "Master Workers" (See presented analysis above). We study how accuracy is impacted varying the number of annotators for a single video. We conduct an experiment to evaluate our agglomerative clustering algorithm. The algorithm is constrained using the overlap between pairs of annotations. Figure 3.8 reports the F-score varying the number of turkers used to annotate each video. Results suggest that selecting around six workers, we can achieve a high quality temporal localization. Moreover, we investigate the effect of the pairwise overlap constrain and report the F-score in Figure 3.9. We note that fixing the overlap

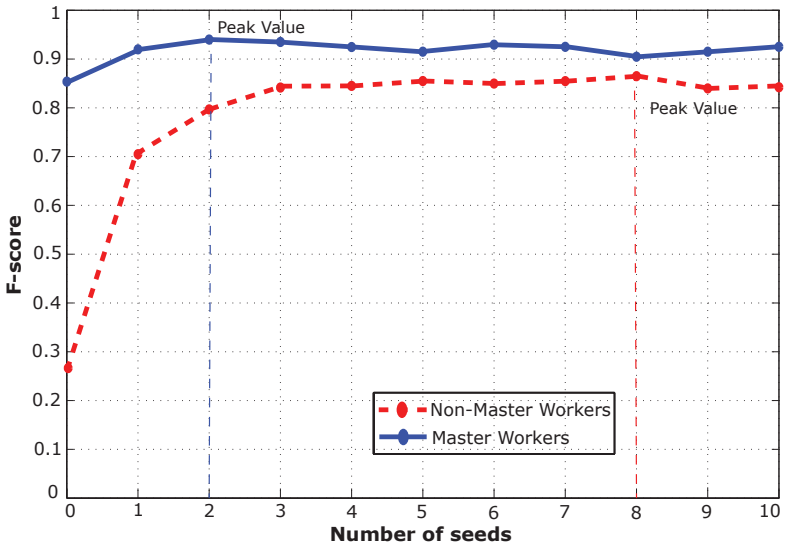

FIGURE 3.7: Comparison of F-score obtained using Non-Master Workers Vs Master Workers. Our experiments exposes that Master Workers perform a more accurate annotations in comparison to Non-Master Workers. We attribute this improvement to that Master Workers requires a large reputation in Amazon Mechanical Turk and could filter out malicious workers.

to 0.6 brings a maximum F-score. This experiment allows the characterization of our agglomerative clustering. We find that F-score decrease when using overlap threshold values near to one due to the strict merge annotation condition. Relaxing the overlap to 0.6 bring us a trade-off between strictness and possibility to join different worker annotations in a same group.

3.2.2 Collection accuracy

3.2.2.1 Benchmark datasets

Olympic sports The Olympic sports dataset [46] contains 16 different sport action classes. It is collected from YouTube and annotated with the help of Amazon Mechanical Turk. This dataset in total contains 783 videos and train/test set split are recommended by the authors. In our experiments, we use the original not trimmed videos in order to evaluate our temporal localization method.

Daily Human Activities We collect a novel dataset of high level human activities. Instead of use turkers, we hire strong supervised annotators to label activities in the videos (in the same pipeline proposed by our framework). Dataset is introduced above in Section 3.2.1.

3.2.2.2 Filtering evaluation

In order to evaluate our filtering stage, we design an experiment in which two datasets are benchmarked. We launch tasks to AMT for all videos in the dataset and compare the filtering results with the Ground Truth. Additionally, we include noisy videos from YouTube with non-related content. We fix the amount of Ground Truth videos around 10% of total videos verified. For each task, we introduce five Gold Standard videos. To perform the task, workers must be qualified as "Master Workers".

We report in Table 3.1 obtained accuracy in benchmark data sets. We observe that our filtering strategy performance not decrease significantly when the number of videos to verify increase. In the Daily Activities dataset the total videos are twice in comparison with the Olympic Sports dataset and the accuracy slightly decrease in 2%.

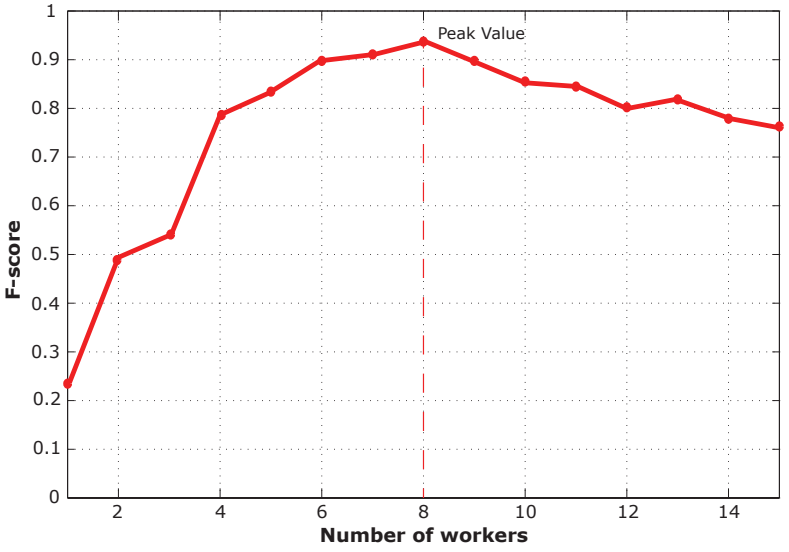

FIGURE 3.8: F-score obtained for different number of workers to annotate each video. Since our temporal localization requires multiple workers to annotate a single video, we study how many workers are needed to annotate each video accurately. To conduct the experiment, we vary the number of workers in a range of [1-10] and find that 8 workers to annotate each video gives us the highest accuracy.

Dataset	Num. HITs	Prec	Rec	F1-score
Olympic Sports [46]	233	0.95	0.93	0.94
Daily Activities	500	0.94	0.89	0.91

TABLE 3.1: Experimental results of filtering candidate videos for two benchmarked datasets. Filtering candidate videos is evaluated as a retrieval problem. Ground Truth consists of a set of accurate annotations by vision researchers, which determine if a video match a desired activity. We make a binary comparison between turkers annotations and Ground Truth to compute precision and recall in retrieved videos.

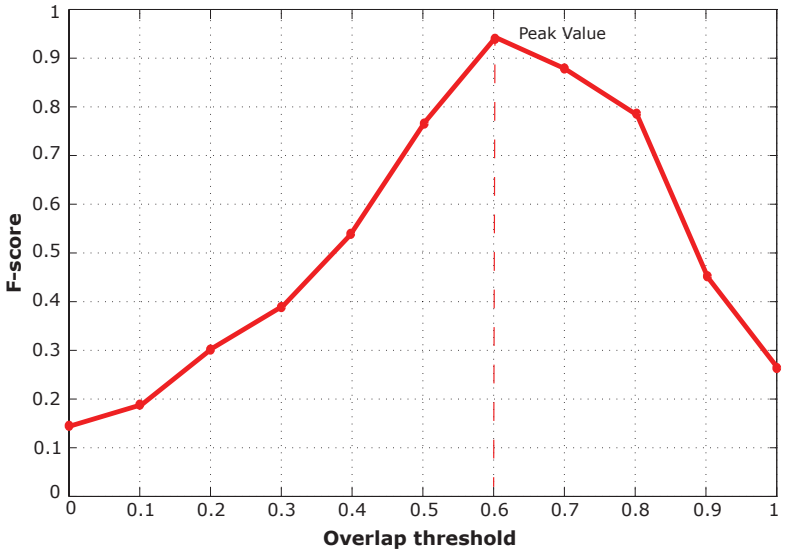

FIGURE 3.9: Overlap threshold analysis. Our agglomerative clustering algorithm requires an overlap threshold constrain as input. We investigate which threshold give a better performance in the evaluated sub-set. Results reveals that fix the overlap threshold to 0.6 provides the more accurate annotations.

3.2.2.3 Temporal localization evaluation

To evaluate our temporal localization approach, we compare turker annotations to Ground Truth. We compute F1-score for the two datasets introduced above. We launch HITs for all videos in the datasets, and hire seven different turkers to annotate starting and ending times of each of these videos. Only "Master Workers" can complete the HIT for temporal localization.

Table 3.2 reports accuracy for temporal localization. We observe that turkers perform annotations with high precision. Recall value is slightly minor in both datasets compared with achieved precision. We attribute this to the large amount of activities that can be found in both datasets. We show that our approach to annotate starting and ending frames allows with high accuracy localize human activities in video sequences. Figure 3.10 shows both successes and failures results, from which we can see that our approach produces reliable annotations.

3.2.3 Large scale human activity database

We construct a novel large scale video database of natural human activities with our proposed framework (see Section 3.1). Our novel database has the following key properties: large scale, high accuracy labels, taxonomy and high diversity. Moreover, we focus

on the collection of user generated videos (i.e. avoiding collect surveillance video data). We provide human annotated temporal locations of activities in each video.

Our novel database contains four top levels type of activities organized : household activities, work activities, personal care and sports. We construct our natural human activity database upon 75 categories of our natural human activities. The hierarchy of the database is four levels deep, ranging from top categories such as household activities to leaf categories such as gardening. Figure 3.11 shows a snapshot of each leaf category and how it is distributed among top levels activities. In the first release of this database, we focus on collecting household activities. However, we plan enlarge the database to more categories. This will be possible due our scalable framework for collecting and annotating human activities in web video data.

Our database contains a rich meta-data provided by YouTube. We provide information such as: video description, tags, author name, title name and user comments. These information could be useful for integrating computer vision algorithms with natural language processing. Furthermore, audio data is also provided. Our entire provided data bring the possibility of develop multimodal systems that can exploit the benefit of these rich data as Fu et al. [18] explore.

We include 4600 videos depicting different natural human activities. Each video contains an accurate annotations of temporal boundaries of activities. Note that our database provide the long sequence in which an small amount of the video is related with the activity. This opens opportunities for the development of automatic trimming or detection activities in long video sequences.

3.3 Evaluation of human activity recognition approaches

In this section, we study the performance of two well established algorithms for action recognition in our novel dataset. First, we define our experimental setup giving a brief

Dataset	Num. HITs	Prec	Rec	F1-score
Olympic Sports	466	0.93	0.90	0.91
Daily Activities	1000	0.98	0.89	0.93

TABLE 3.2: Temporal localization results on benchmarked datasets. Ground Truth is composed by expert annotations (vision researchers) of starting and ending times of human activities on non-trimmed videos. We compute the overlap between (intersection size divided by union size) all pairs of turkers annotations and Ground Truth data. If obtained overlap coefficient is greater than 0.6, we count the annotation as a true positive. Accuracy is measured by precision and recall values.

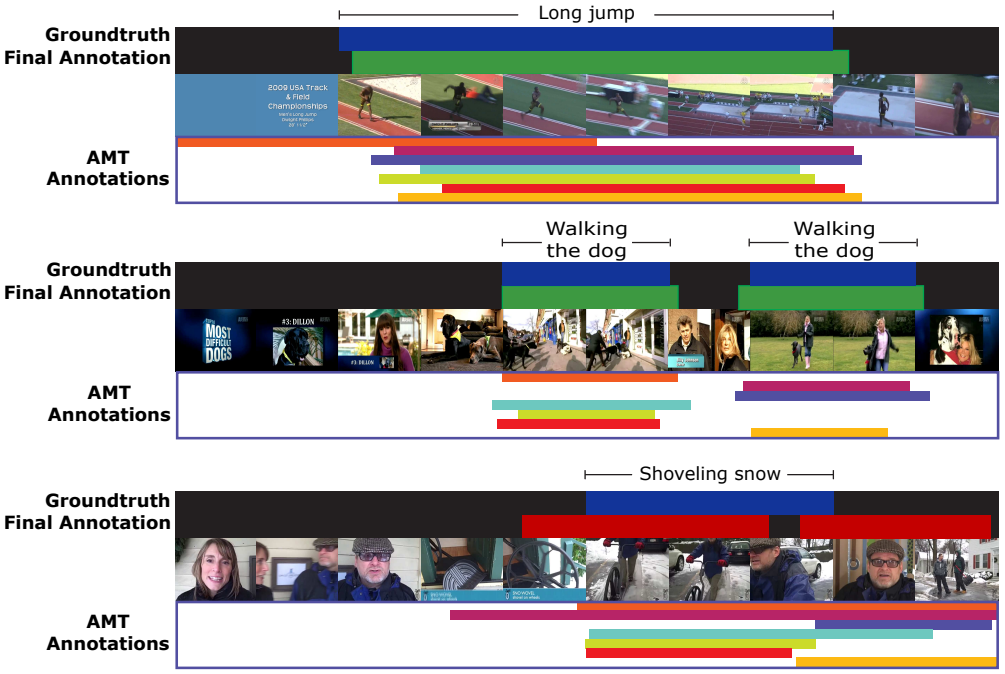

FIGURE 3.10: Successful and failures annotation results. We illustrate annotation results for long jump (Olympic Sports), walking the dog (from our novel dataset) and shoveling snow (from our novel) activities. Each sub-figure shows annotations performed by AMT workers (bottom), the final annotation obtained with our agglomerative clustering algorithm (bar at top of video frames), Ground Truth (at top of final annotation) and ten video frames sampled from the originals videos. **Top** sub-figure shows an example annotation result for long jump activity. Our method accurately annotate starting and ending times of the mentioned activity. We observe in this example, that the first AMT annotation is not reliable, then we consider it as a bad annotation (worker is not rewarded). **Center** sub-figure presents an example where two instance of walking the dog activity are annotated accurately. **Bottom** sub-figure illustrates a bad result for shoveling snow activity. We observe that turker annotations has several distributions among the video. We attribute this behavior to the fact that the video contain several changes in camera view points and annotate the starting and ending frames becomes a subjective task.

description of the methods implemented. Next, we discuss the recognition results of these approaches.

3.3.1 Experimental setup

We implement two popular approaches for action classification [34] and [61] to evaluate their recognition performance in our novel dataset of daily activities. Both methods rely on the standard Bag of Features representation for videos. To recognize actions, a Support Vector Machine (SVM) is used for learning models for each class in our dataset.

FIGURE 3.11: Large scale natural human activities video database. The figure shows a snapshot of each leaf category and how it is distributed among top levels activities. In the first release of this database, we focus on collecting household activities.

In this sense, the methods differ only on the feature extraction stage. Moreover, we investigate the recognition performance of static spatial features such as SIFT [38] in the same pipeline.

The method in [34], detects spatio-temporal interest points using [33]. They compute Histogram of oriented gradients (HOG) and Histograms of flow (HOF) descriptors upon

a neighborhood of detected points. Then, a codebook of visual words is constructed for each descriptor type. Finally, for each video, all features are quantized according to the computed codebook and the resulting histograms are used to train action models using a SVM.

Wan et al. [61], implement a different feature extraction approach. Their method computes several descriptors (HOG, HOF, Motion Boundaries Histogram, Trajectory shape) upon a dense set of trajectories. These trajectories are computed using optical flow to track dense points in all frames in the video sequence. They follow a similar pipeline such as used in [33] for video representation and to learn action models (bag of word approach along with a SVM approach).

Finally we obtain the accuracy in order to measure the performance for each implemented method. We consider the most commonly used criterion [33, 34, 61] to measure classification accuracy: model result matches a binary Ground Truth and count as a (1) true positive, (2) true negative, (3) false positive and (4) false negative. Finally, accuracy is computed in an one-vs-all strategy as in [33, 61]

3.3.2 Classification results

Figure 3.12 shows the accuracy obtained by four different approaches to recognize activities in our novel dataset. The Multi-channel approach, implements a similar strategy as in [67] to combine all descriptors of each tested method.

We observe that dense trajectories [61] achieve better results than STIP [34] and SIFT [38] approaches. However when descriptors for each method are combined (Multi-channel approach), performance recognition significantly increase. We attribute this improvement to the ability of the Multi-channel approach to merge each descriptor visual contribution. SIFT [38] captures information about the scene where the activity occurs. In other hand, STIP [34] and the dense trajectories[61] capture a rich information for both visual appearance of actions (HOG descriptor) and motions involved in the execution of the activity (HOF, MBH).

We present classification results for the Multi-channel approach in Figure 3.13. We observe that the confusion matrix has an strong diagonal (performance = 69.9%). Nevertheless, we find that our novel dataset represent a challenge for the tested approaches due to the high variety contained in the dataset. We note that a large amount of videos have several camera motions that includes noisy information into descriptors. In order to overcome this challenge, we introduce a method to stabilize feature trajectories in Chapter 4.

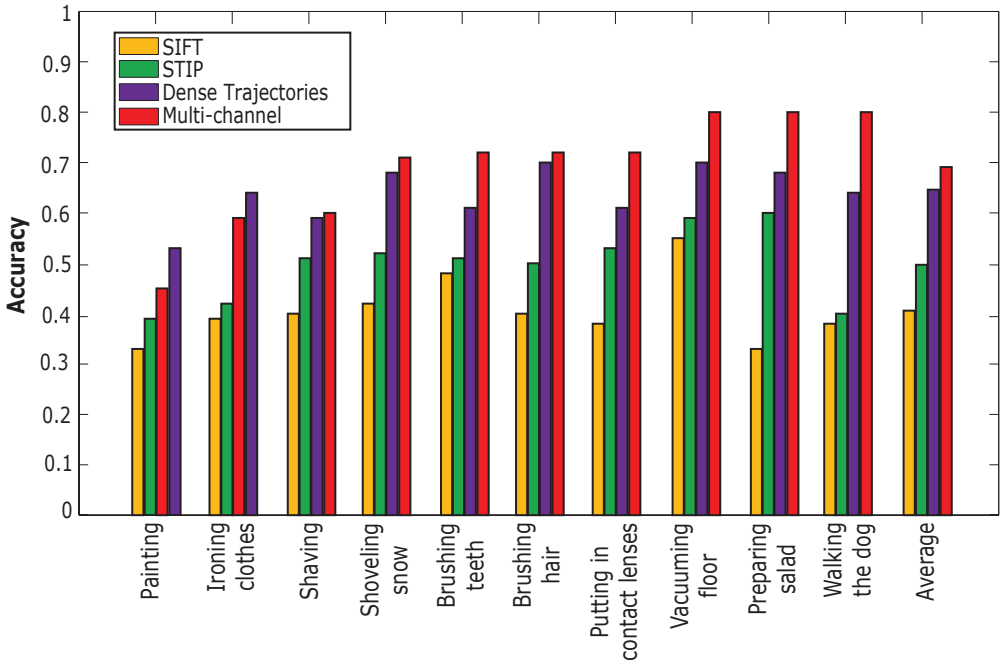

FIGURE 3.12: Performance comparison of different approaches for action recognition. We evaluate four different approaches in our novel dataset of daily activities. The Multi-channel approach achieves the better results in our benchmark. We attribute this to the inclusion of several types of description (scene, spatio-temporal appearance, spatio-temporal motions).

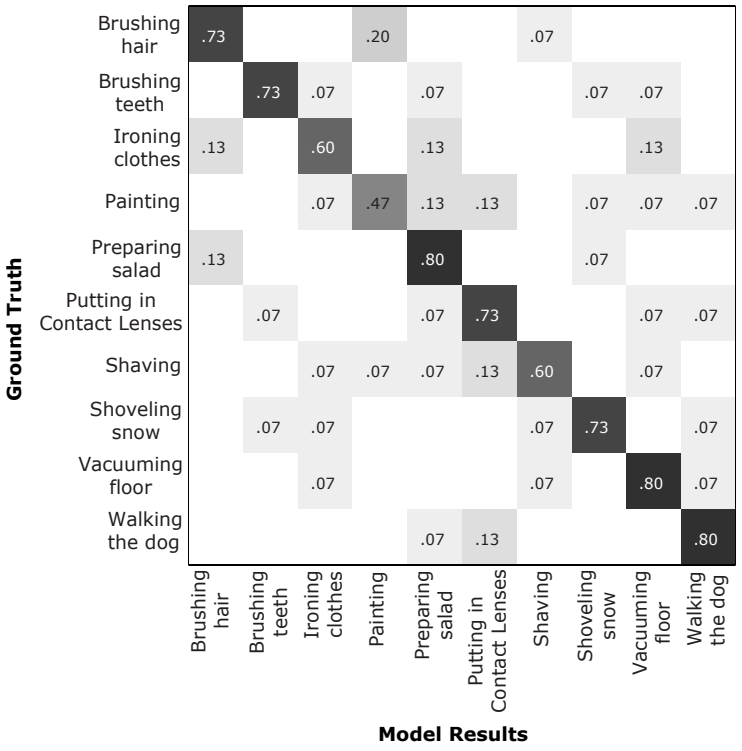

FIGURE 3.13: Confusion matrix for Multi-channel approach (performance average = 69.9%).

Chapter 4

Stabilized Trajectory Feature for Human Activity Recognition

Human action recognition is a challenging task for computer vision algorithms due to large variability in video data caused by occlusions, camera motions, actor and scene appearances, among others. We introduce in Chapter 3 a novel dataset of human activities. We find that many videos has several camera motions inducing strong noises to motion descriptors. In this chapter, we study different approaches for robust feature extraction and propose a novel algorithm capable for overcoming the abrupt camera motions in videos.

A popular current trend in action recognition methods relies on using local video descriptors to represent visual events in videos [13, 33, 61]. These features are usually aggregated into a compact representation, most commonly into a Bag of Features (BoF) representation framework [51]. The advantage of this simple representation is that it avoids difficult pre-processing steps such as motion segmentation and tracking. In the BoF representation, local descriptors are quantized using a pre-computed codebook of visual patterns. This representation combined with discriminative classifiers such as Support Vector Machines, has achieved tremendous success in action recognition in controlled scenarios [21, 51]. Due to its simplicity, BoF requires the use of strong, robust and informative features, which can be obtained reliably in such simplified scenarios. However, recent efforts in the collection of more realistic datasets from movies and web sites [32, 37, 40] represent a challenge for existing methods due to dynamic backgrounds, changes in light conditions and camera motions among other noisy conditions.

In order to overcome the challenges of realistic datasets, Wang et al. [61] introduce the use of dense trajectories for action recognition. Their method densely samples feature points that are tracked over a fixed time span using optical flow. Once trajectories

FIGURE 4.1: Trajectory features in non-stabilized and stabilized video. In this chapter, we introduce a method for computing point trajectory features on videos acquired with moving cameras. Our method computes a weak video stabilization that eliminates coarse motion while preserving subtle visual motions of interest. **Top**: Original video. **Bottom**: Stabilized video. **Left**: Overlay of frames t and $t + 5$. **Right**: Extracted trajectory features.

are located in the video, their algorithm computes local descriptors around each video neighborhood to capture texture and motion patterns. Unfortunately, trajectories are heavily corrupted when there is large camera motion and dynamic backgrounds (See top row in Figure 4.1).

In this chapter, we propose a method for trajectory stabilization that overcomes undesired camera motions while preserving the subtle motions of interest related to the events in the video. Our algorithm performs weak video stabilization in order to filter out background feature points and to compensate for coarse motion when computing motion descriptors. We evaluate the performance of our feature stabilization approach in three current benchmarking datasets: a)**Hollywood 2** [40]: a movie-centric action dataset that incorporates challenging camera view-point changes and professional camera movements; b)**HMDB51** [32]: a dataset of human actions in videos retrieved from the web, which contain large and undesired camera motions; c)**Olympic Sports** [46]: a sport related database with videos retrieved from the web. Additionally, we evaluate the performance in our novel dataset of natural human activities. We evaluate the effectiveness of our stabilized trajectory features by measuring recognition performance when combined with several local descriptors under the simple BoF assumption.

4.1 Feature stabilization approach

In this section, we introduce our approach to compute robust trajectories features for visual description of human actions. We present a summary of our feature stabilization method in Figure 4.2. Given an input video we first compute a coarse optical flow. We then perform weak video stabilization by using the flow vectors to align pairs of consecutive video frames. We track feature point and extract trajectories from the weakly stabilized video. Finally, we compute local video descriptors around the neighborhood of each extracted trajectory.

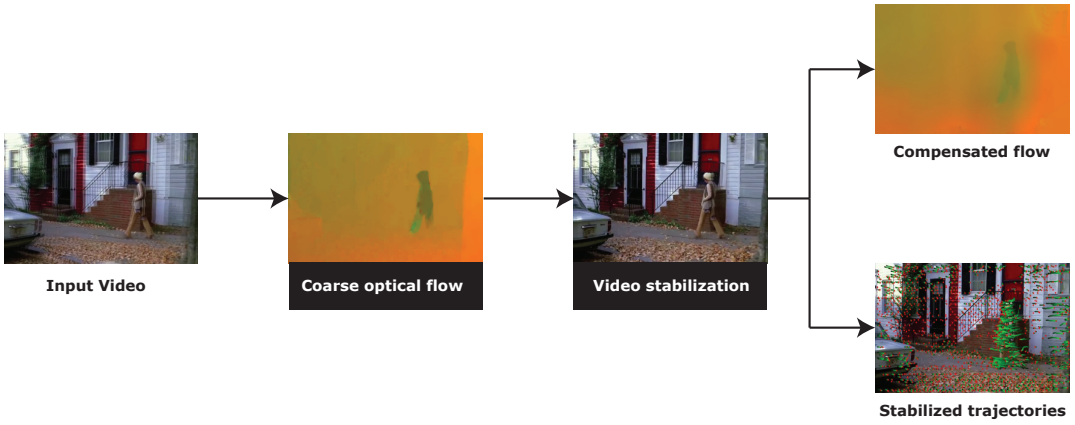

FIGURE 4.2: Pipeline of the feature stabilization approach. The first step is to compute a coarse optical flow that preserve motions related with the events in the video. Then, a video stabilization is performed aligning pair frames with the coarse optical flow. Next, stabilized feature trajectories are computed and finally different local descriptors are computed

4.1.1 Weak stabilization

The goal of video stabilization is to eliminate coarse motions while isolating subtle motions that better represent the event of interest. Instead of performing feature-based alignment, we compute a dense but coarse optical flow between consecutive video frames and warp images using the flow vectors. Our stabilization methodology is inspired by [47], where weak stabilization is used to eliminate pedestrian translations while preserving limb movements.

In our framework, we compute coarse optical flow using the method of Lucas-Kanade [39]. In order to extract only coarse motion, we use a large window radius σ, which effectively smooths small motion details and preserves coarse, large motion.

To stabilize video, we align pairs of frames using the coarse flow vectors [47]. Figure **??** shows an example result of our weak video stabilization. We observe that the method

performs well and is able to stabilize both camera and large object centric motions. Some failure cases may occur when there are very large displacements between consecutive frames.

4.1.2 Stabilized trajectory feature extraction

Once weak stabilization is performed, our method extracts stabilized feature trajectories. We follow the dense trajectory pipeline introduced by Wang et al. [61]. We sample feature points densely in several spatial scales and track them for a fixed amount of frames. We can then remove static trajectories that do not change their location through time in the stabilized video. Note that we also extract local descriptors from the stabilized video, so that motion descriptors also capture the residual motion information after the coarse motion has been eliminated.

In the Figure 4.3 we present results of our stabilized features on videos with different types of camera movements. First row shows an example of shaking camera movement. We observe that a set of features are generated by the camera motion. These trajectories are combined with the foreground features, so that the signal to noise ration is rather low, yielding inaccurate visual descriptors.

Otherwise, our stabilized features handle well the camera motion, removing most trajectories related with the background. The other two examples shows a more critical effect without video stabilization. With our algorithm, the feature stabilization process can remove most non-related trajectories while preserving trajectories related to human movements.

4.1.3 Implementation details

Instead of stabilizing the entire sequence, our weak stabilization implementation operates on local time ranges using a temporal sliding window of length L. We perform weak stabilization only across the L frames in each temporal window. Local stabilization is important to preserve shapes and avoids deformation artifacts. We compute the global motion summing and warping flow fields progressively as is done in [47]. In our experiments, we fixed L to the same length of trajectories ($L = 15$). We set the value of σ to 32. We investigate the effects of these parameters in by measuring the performance of our action recognition pipeline. Experimental evaluation of parameter sensitivity are reported in Section 4.2.3.

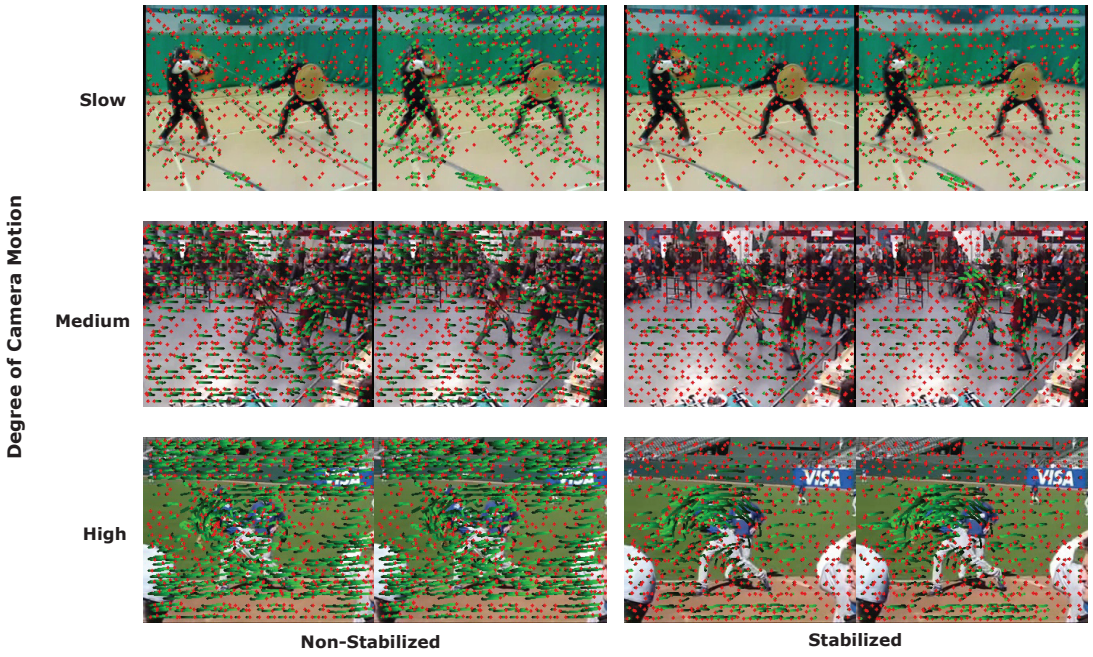

FIGURE 4.3: Comparison between stabilized and non-stabilized trajectory features. Trajectory features obtained from Non-Stabilized and Stabilized video. Examples of different types of camera motion shows that stabilized features are more related with the human motion.

4.2 Experimental evaluation

4.2.1 Experimental setup

In this section, we describe our experimental setting. First, we introduce the four datasets used in our evaluations [32, 40, 46] which are the most challenging due to their large scale and realistic conditions. Second, we detail the classification setup.

4.2.1.1 Datasets

Hollywood2 The Hollywood2 dataset [40] contains a large collections of videos retrieved from 69 different Hollywood movies divided in 12 action classes. It contains a set for training of 823 videos and a test set of 884 videos. This dataset is challenging due to several professional motion camera (i.e. pan, tilt, dolly). To evaluate the performance, we follow the protocol established in [40] measuring the mean average precision (mAP) over all classes.

Olympic sports The olympic sports dataset [46] is defined above in Section . In our experiments, we report as recognition performance the mAP as proposed by [46].

HMDB51 The HMDB51 dataset [32] is a large video collection of human actions (51 action classes). It contains 6766 videos from different sources ranging from digitalized movies to user-generated YouTube videos. Three different train/test splits are provided by the authors. We measure recognition performance by the average accuracy over the three suggested splits by authors.

Daily human activities Our novel dataset of natural high level human activities. The dataset includes 10 different classes of daily human activities (see Section 3.2.1 for more details). We measure recognition performance with the average accuracy over ten different splits upon a k-fold strategy.

4.2.1.2 Classification setup

Given the dense trajectories features, we train a codebook for each descriptor using the k-means algorithm. To construct the codebook, we sample 100,000 random features points and set codebook size to 4,000. We then quantize descriptors to their closest codebook word. Therefore, resulting histograms are used as the feature vector for classification. In order to recognize actions, we learn a non-linear SVM with χ^2 kernel. We combine different descriptors using a Multi-channel approach as in [67]:

$$K(x_i, x_j) = \exp(-\sum_c \frac{1}{\Omega_c} D_c(x_i, x_j)) \tag{4.1}$$

where, $D_c(x_i, x_j)$ is the χ^2 distance for channel c, and Ω_c is the average channel distance.

4.2.2 Stabilized feature trajectories

To evaluate our stabilized feature trajectories, we use the four benchmarks cited above. We compare recognition performance for each descriptor type (Trajectory, HOG, HOF, MBH) with and without our stabilization approach. We report in Table 4.1 obtained results for both [61] (our baseline) and for our feature stabilization approach. In all benchmarked datasets we outperform the recognition performance reported by [61]. We attribute this improvement to the ability of our method to suppress coarse motions while preserving the subtle motions of interest related to the events in the video.

We find that motion descriptors are enhanced with our feature stabilization. First, HOF descriptors increment significantly their performance in all evaluated datasets. HOF improvements reaches up to 11.4% in Olympic sports. Moreover, we observe that MBH descriptors are also improved with our stabilized features. We attribute this

	Hollywood2		Olympic Sports		HMDB51		Daily Human Activities	
Descriptor	[61]	Ours	[61]	Ours	[61]	Ours	[61]	Ours
Trajectory	47.8%	41.8%	60.5%	75.3%	28.0%	30.5%	59.3%	61.2%
HOG	41.2%	46.1%	63.0%	76.2%	27.9%	39.1%	62.7%	62.5%
HOF	50.3%	57.3%	58.7%	83.2%	31.5%	47.5%	63.8%	69.1%
MBHx	N/A	60.1%	N/A	86.5%	N/A	53.1%	N/A	69.4%
MBHy	N/A	61.3%	N/A	**86.9%**	N/A	52.8%	N/A	69.1%
MBH	55.1%	60.5%	71.6%	85.7%	43.2%	52.4%	63.9%	69.5%
All descriptors	58.2%	**63.5%**	74.1%	**86.3%**	46.6%	**54.3%**	64.1%	**69.7%**

TABLE 4.1: Comparison of recognition performance for local descriptors. Performance in all motion descriptors (Trajectory, HOF, MBH) are improved with ours stabilized feature trajectories. HOG descriptor is slightly improved due to point features are more related with the events.

improvement to the ability of our approach to capture large camera motions adding a more robust description to the MBH first derivatives. Otherwise, the Trajectory descriptor is only slightly improved in Hollywood2 dataset, but there is an improvement around 10% in recognition performance in Olympic Sports and HMDB51.

In Table 4.2, we compare our method with state of the art methods for action recognition capably to handle camera motion. We report performance recognition for different approaches [27, 28, 31, 50, 56, 61]. Most related to our approach, Dense Trajectories [61] performance is reported due to the simplicity of the method and its extensive popularity in the literature. Closely related to our method, ω-flow (Jain et al. [27]) is compared and results are also reported in Table 4.2.

We observe that our approach achieves better recognition performance than Dense Trajectories [61] in all benchmarked datasets. We outperform Dense Trajectories [61] due to their HOF descriptor is computed with an inaccurate feature points (see Table 4.1). Moreover, we observe that filtering background trajectories can improve HOG descriptor around five percent in each dataset as reported in Table 4.1. Otherwise, we attribute the better recognition performance achieved compared to Jain et al. [27] due to our stabilized features are stronger compared to their residual motion, which has problems to tackle with videos where the dominant motion is related with the actor. Table 4.2 shows that we outperform ω-flow in three experimental benchmarks. We improve their results up to 3.1% in Olympic sports.

We note that feature stabilization significantly benefit the recognition performance in our novel dataset of natural human activities. We attribute this to that when stabilize features, a better motion description is obtained due to the correction of the flow field. Since a large amount of video of this dataset contains large camera movements, our

Method	Hollywood2	Olympic Sports	HMDB 51	Daily Human Activities
Saliency [56]	60.0%	N/A	N/A	N/A
Action Bank [50]	N/A	N/A	26.9%	N/A
MIP [31]	N/A	N/A	29.17%	N/A
Dense Trajectories [61]	58.2%	74.1%	46.6%	64.1%
Reference Points [28]	59.5%	80.6%	40.7%	N/A
ω-flow [27]	62.5%	83.2%	52.1%	N/A
Our method	**63.5%**	**86.3%**	**54.3%**	**69.7%**

TABLE 4.2: Comparison of different action recognition methods using robust feature extraction.

stabilization approach brings a more robust way to compute the HOF, MBH and Trajectory descriptors. Moreover, our approach tries to focus on foreground motion filtering the coarse motions which tends to be the camera motion.

4.2.3 Parameter study

In this section we evaluate recognition performance in Olympic sports dataset. We investigate the effect of the parameter σ in conjunction with different stabilization window length L. Figure 4.4 shows performance recognition with different pairs of values for σ and L. We observe that large σ benefit action recognition due to preserve human part motions after align coarse motions. When we set σ to 8, performance recognition decrease around 8%, which is a similar result obtained by the baseline [61]. Additionally, we study how performance recognition is affected by L. We note that large values of σ combined with short trajectories achieve the better performance in our recognition setting. Finally, we observe that $\sigma = 32$ and $L = 15$ gives a well performance as Figure 4.4 illustrate. Our experimental results show that computing a dense but optical coarse flow improves the performance recognition due to feature trajectories becomes reliable in the human motions.

4.2.4 Computational cost

In this section we evaluate the computational complexity of our feature stabilization. We compare with the two different methods described in Section 4.2.2. We measure the frames per second (FPS) processed by each method. We select a video with resolution of 480x352 and extract features for each method. We obtain the runtime in a Dell work station with a Four Core XEON (E5-1620) and 16GB RAM. Table 4.3 shows the FPS spent for the different methods. Our approach obtains an slow computation as a result of the additional computation of coarse optical flow for feature stabilization. However,

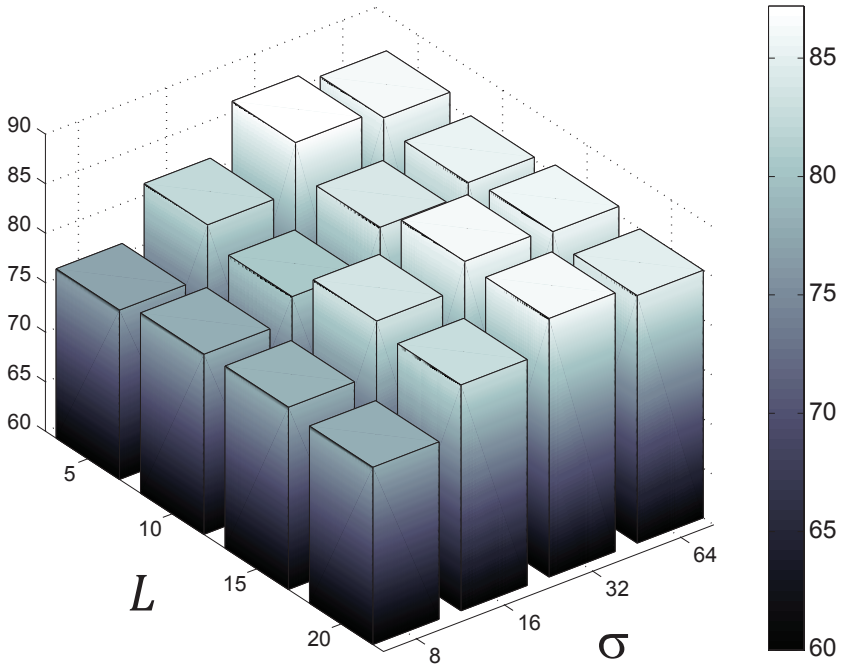

FIGURE 4.4: Evaluation of stabilized features trajectories parameters. Experiments are conducted in Olympic sports dataset. We report the performance recognition varying σ and L. Experiments results shows that large values of σ benefit the overall performance recognition.

Method	FPS
DenseTrajectories [61]	3.53
ω-flow [27]	2.10
Our method	2.73

TABLE 4.3: Computational cost comparison between several feature extraction methods. We achieve a slightly gain in computational cost compared to ω-flow method.

uur method is slightly faster than ω-flow [27] (state-of-the-art) due to we avoid the Affine Model computation to estimate camera motion.

Chapter 5

Conclusion

In this work, we proposed an approach to automatically search and annotate human activities. We have empirically shown that our collection/annotation framework allows high quality labeling for both filtering candidate videos and annotating starting and ending times of human activities in video sequences. We built a large scale video database of natural human activities using the proposed annotation framework. Additionally, we introduced a method to stabilize feature trajectories. We demonstrated that our stabilization approach improved human action recognition in videos with camera movements as a result of the trajectory filtering and flow field correction. Moreover, we showed that our method benefit significantly the performance recognition of visual motion descriptors.

One possible future research direction would insert user-generated natural language descriptions and construct a hierarchical structure based on micro-actions that compose the high level activities to obtain a deeper knowledge from the video data. We plan to enlarge our database of natural human activities with annotations at multiple scales. The database extension could provide spatio-temporal bounding boxes of each taxonomic activity; we would also annotate region-based finer level micro-actions or motions. Moreover, we expect to formulate a novel evaluation methodology that explode the hierarchy and knowledge structure of our database.

Glosary

Annotate. Manually temporal tagging of human activities.

Atomic action. Human activity which only require simple human movements (running, jumping, etc).

Crowdsourcing. Business model in which are outsourced to an undefined large group of people in the form of an open call.

Gold Standard. Refers to ground truth data which are introduced in a crowdsourced task in order to detect reliable workers.

Ground Truth. Data used to evaluate supervised machine learning algorithms.

High Level Activity. Human activity that involve interaction with other people, objects or animals.

HIT. Refers to a Human Intelligent Task performed by Amazon Mechanical Turk workers.

K-fold. Cross-validation technique which randomly split data set into K equal size.

Master Workers. Masters are elite groups of Workers who have demonstrated accuracy on specific types of HITs on the Mechanical Turk marketplace.

Bibliography

[1] J. D. Abernethy and R. M. Frongillo. A collaborative mechanism for crowdsourcing prediction problems. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2600–2608. 2011.

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis. *ACM Computing Surveys*, 43(3):1–43, Apr. 2011.

[3] K. Ali, D. Hasler, and F. Fleuret. Flowboost: Appearance learning from sparsely annotated video. In *CVPR*, 2011.

[4] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[5] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[6] M. Brown and D. G. Lowe. Recognising panoramas. In *ICCV*, volume 3, page 1218, 2003.

[7] C. Buehler, M. Bosse, and L. McMillan. Non-metric image-based rendering for video stabilization. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–609. IEEE, 2001.

[8] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 885–891. IEEE, 1998.

[9] A. Censi, A. Fusiello, and V. Roberto. Image stabilization by features tracking. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 665–667. IEEE, 1999.

[10] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[11] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popović, et al. Predicting protein structures with a multiplayer online

game. *Nature*, 466(7307):756–760, 2010.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[13] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior Recognition via Sparse Spatio-Temporal Features. In *VSPETS*, 2005.

[14] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009.

[15] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733. IEEE, 2003.

[16] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid models for human motion recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 1166–1173. IEEE, 2005.

[17] R. B. Fisher. The pets04 surveillance ground-truth data sets. *Proc. 6th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2004.

[18] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Attribute learning for understanding unstructured social activity. In *Computer Vision–ECCV 2012*, pages 530–543. Springer, 2012.

[19] M. L. Gleicher and F. Liu. Re-cinematography: Improving the camerawork of casual video. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 5(1):2, 2008.

[20] R. G. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 558–566. 2011.

[21] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[22] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 225–232. IEEE, 2011.

[23] M. Hansen, P. Anandan, K. Dana, G. Van der Wal, and P. Burt. Real-time scene stabilization and mosaic construction. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 54–62. IEEE, 1994.

[24] J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[25] P. Ipeirotis, F. Provost, V. Sheng, and J. Wang. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, 2013.

[26] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on Human Computation (HCOMP 2010)*, pages 64–67. ACM, 2010.

[27] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR - International Conference on Computer Vision and Pattern Recognition*, Portland, États-Unis, Apr. 2013.

[28] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ECCV 2012*, volume 7576 of *Lecture Notes in Computer Science*, pages 425–438. Springer Berlin Heidelberg, 2012.

[29] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1953–1961. 2011.

[30] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

[31] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *European Conference on Computer Vision (ECCV)*, Oct. 2012.

[32] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011.

[33] I. Laptev. On Space-Time Interest Points. *IJCV*, 64(2-3):107–123, 2005.

[34] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *Spatial Coherence for Visual Motion Analysis*, pages 91–103. Springer, 2006.

[35] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[36] H. Liu, R. Feris, and M. Sun. Benchmarking human activity recognition. *CVPR Tutorial, CVPR*, 2012.

[37] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1996–2003. IEEE, 2009.

[38] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[39] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[40] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2929–2936. IEEE, 2009.

[41] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 514–521. IEEE, 2009.

[42] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(7):1150–1163, 2006.

[43] D. Mihalcik and D. Doermann. The design and implementation of viper. Technical report, 2003.

[44] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[45] L.-V. Nguyen-Dinh, C. Waldburger, D. Roggen, and G. Tröster. Tagging human activities in video by crowdsourcing. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR 2013)*, Apr. 2013.

[46] J. C. Niebles, C.-W. Chen, , and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[47] D. Park, C. Zitnick, D. Ramanan, and P. Dollár. Exploring Weak Stabilization for Motion Feature Extraction. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2013.

[48] V. C. Raykar and S. Yu. Ranking annotators for crowdsourced labeling tasks. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1809–1817. 2011.

[49] M. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[50] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1234–1241. IEEE, 2012.

[51] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004.

[52] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[53] C. Sun, I. Junejo, and H. Foroosh. Action recognition using rank-1 approximation of joint self-similarity volume. In *Computer Vision (ICCV), 2011 IEEE International Conference on*.

[54] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[55] P. Venetis and H. Garcia-Molina. Quality control for comparison microtasks. In *Proceedings of the First International Workshop on Crowdsourcing and Data Mining*, pages 15–21. ACM, 2012.

[56] E. Vig, M. Dorr, and D. Cox. Space-variant descriptor sampling for action recognition based on saliency and eye movements. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ECCV 2012*, volume 7578 of *Lecture Notes in Computer Science*, pages 84–97. Springer Berlin Heidelberg, 2012.

[57] S. Vijayanarasimhan, P. Jain, and K. Grauman. Far-sighted active learning on a budget for image and video recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3035–3042. IEEE, 2010.

[58] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[59] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 2013.

[60] C. Wah. Crowdsourcing and its applications in computer vision. *University of California, San Diego*, 2006.

[61] H. Wang, A. Klser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.

[62] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–120. IEEE, 2004.

[63] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1419–1426. IEEE, 2011.

[64] S. Yonemoto. A video annotation tool using vision-based ar technology. In *Cyberworlds (CW), 2012 International Conference on*, pages 226–230, 2012.

[65] YouTube. Youtube statistics.

[66] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme video: building a video database with human annotations. In *ICCV*, 2009.

[67] J. Zhang, M. Marszaek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

[68] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. In *Computer Vision and Pattern Recognition, 1997.*